/* **Program6:** Design, Develop and Implement a menu driven Program in C for the following operations on Double Ended QUEUE of integers (Array Implementation of Queue with maximum size MAX)
a. Perform Insertion / Deletion at front of QUEUE
b. Perform Insertion / Deletion at rear of QUEUE
c. Display the status of Circular QUEUE
d. Exit
Support the program with appropriate functions for each of the above operations. */

```
#include<stdio.h>
#include<stdlib.h>
#define MAX 5
int q[MAX];
int front = -1, rear= -1;

void insertAtRear(int item)
{
        if ( rear == MAX - 1 )
 {
  printf ( "\nDequeue is overflow" );
 }
 else
       {
              if(front==-1 &&rear==-1)
                        front=rear=0;
              else
                         rear=rear+1;
              q[rear] = item;
        }
}

void deleteAtFront()
{
          int item;
          if(front==-1)
                      printf("\nDequeue underflow");
          else
          {
                     item = q[front];
                     printf("\nItem deleted from queue is %d",item);
                     if(front == rear)
                               front = rear = -1;
                     else
                               front = front+1;
           }
}

void insertAtFront(int item)
{
        int i;
        if( front == 0 && rear == MAX - 1 )
        {
                printf ( "\nDequeue is overflow" ) ;
        }
        else
        {
                 if(front == -1 &&rear == -1)
                 {
                          front = rear = 0;
                 }
                 else if(front == 0 && rear>=0)
                 {
                         for(i = rear; i>=front; i--)
                                     q[i+1] = q[i];
                          rear = rear+1;
                  }
                  else if(front != 0)
                 {
                          front = front - 1;
                 }
                 q[front]=item;
         }
}

void deleteAtRear()
{
          int item;
          if(front==-1)
                     printf("\nDequeue underflow");
          else
          {
                     item = q[rear];
                     printf("\nItem deleted from queue is %d", item);
                     if(front == rear)
                              front = rear = -1;
                     else
                              rear = rear-1 ;
          }
}

void display()
```

```
{
        int i;
        printf("\nfront  = %d  rear=%d", front, rear);
        if(front == -1)
        {
                printf("\nEmpty");
                return;
         }
         else
        {
               printf("\nContents are: \n");
               for(i=front; i<=rear; i++)
                          printf("%d ", q[i]);
        }
}

int main()
{
    int ch, item;
   while(1)
   {
     printf("\n\n~~Menu~~");
     printf("\n1.Insert at front");
     printf("\n2.Delete from front");
     printf("\n3.Insert at rear");
     printf("\n4.Delete at rear");
     printf("\n5.Display");
     printf("\n6.Exit");
     printf("\nPlease enter a valid choice:");
     scanf("%d",&ch);
     switch(ch)
     {
       case 1: printf("\nEnter the item to be inserted:  ");
            scanf("%d", &item);
            insertAtFront(item);
            display();
            break;

       case 2: deleteAtFront();
            display();
            break;

       case 3: printf("\nEnter the item to be inserted:  ");
            scanf("%d",&item);
            insertAtRear(item);
            display();
```

```
                break;

            case 4: deleteAtRear();
                display();
                break;

            case 5: display();
                break;
            case 6: exit(0);
            default: printf("\nEnter the valid choice:  ");
                break;
        }
    }
    return 0;
}
```